JPRC CONFIDENTIAL

COPY

Invention Disclosure Record

Justsystem Pittsburgh Research Center

Method and Apparatus for Efficient Identification of Duplicate and Near-Duplicate Documents and Text Spans using High-Discriminability Text Fragments

June 24, 1999

Mark Kantrowitz

JPRC Invention Disclosure Record

Method and Apparatus for Efficient Identification of Duplicate and Near-Duplicate Documents and Text Spans using High-Discriminability Text Fragments

Abstract

The present invention identifies duplicate and near-duplicate documents and text spans by identifying a small number of distinctive phrases (word n-grams) for each document. The phrases act as a proxy for the full document, allowing the invention to compare documents by comparing their distinctive phrases. Indexes from phrases to document sets and documents to phrase sets allows the present invention to find near-duplicate documents efficiently without needing to compare each pair of documents. The result is a method and apparatus for finding near-duplicate documents in large document collections much faster than might be possible with the state of the art document comparison algorithms. Applications of the present invention include removing redundancy in document collections (including web catalogs and search engines), detection of plagiarism and copyright infringement for text documents and passages, and matching summary sentences with corresponding document sentences. Other applications include detecting copyright infringement of images according to image content without needing to add watermarks or other hidden modifications.

Detailed Description of Invention

The present invention concerns a new efficient method for identifying duplicate and near-duplicate documents in a large collection of documents, but without requiring the comparison of each document with all the other documents.

The intuition behind the present invention is that near-duplicate documents will contain long stretches of identical text in common that are not present in other, non-duplicate documents. This is true not just when the text is excerpted, but also when deliberate changes have been made to the text, ranging from the interspersing of comments by another author and paraphrasing to outright plagiarism. As long as the copy of the document is not completely rewritten, there will still be large text fragments that are specific to the document and its duplicates. On the other hand, the text fragments in common between two non-duplicate documents will likely be in common with many other documents.

So if we can find long text fragments that are present in only a few documents, they may represent good evidence that the documents are near-duplicate. These text fragments represent a kind of "signature" for a document that can be used to match the document with near-duplicate documents and to distinguish the document from non-duplicate documents. Documents that overlap significantly on such text fragments of high-intermediate rarity will most likely be duplicates or near-duplicates. These text fragments

represent distinctive features that can be used to distinguish similar documents from dissimilar documents in a robust fashion.

Given an efficient method for finding the distinctive features, we can identify likely near-duplicate documents by counting the proportion of such features in common between the two documents. Indexes from documents to sets of distinctive features and vice versa allow us to measure the overlap without resorting to pairwise comparison of documents.

The key to the effectiveness of this method is the ability to find the distinctive features. We need features that are rare enough to be common among only near-duplicate documents, but not so rare as to be specific to just one document. Even if individual words are not rare enough, n-grams of such words might be. However, longer n-grams might be too rare, and blindly gathering all n-grams of appropriate rarity would yield a computationally expensive algorithm. The number of distinctive features must be small in order for the algorithm to be computationally efficient. Therefore, the task is to find a method that strikes a balance between appropriate rarity and computational expense. The present invention incorporates several such methods.

We describe the present invention in detail, focusing on the method of finding distinctive phrases. After describing the invention, we discuss its performance, a few variations, and some applications of this technology.

Let DF(x) be the number of documents containing the text "x", and N be the overall number of documents. Let R be a threshold on DF. Possible choices for R include a constant, a fixed percentage of N (e.g., 5% of N), the logarithm of N, or the square root of N.

A first pass over all the documents computes DF(x) for all words in the documents after converting the words to lowercase and removing punctuation from the beginning and end of the word. Optionally, a word in a particular document may be restricted from contributing to DF(x) if the word's frequency in that document falls below a user-specified threshold.

The second pass gathers the distinctive phrases. A phrase consists of words which occur in more than one document (i.e., DF(x) > 1) and in no more than R documents (i.e., DF(x) < R). Phrases may also contain "glue words" which occur in at least N – R documents. Glue words include stopwords like "the" and "of", and allow phrases like "United States <u>of</u> America" to be counted as distinctive phrases. Glue words, which are words for which their absence is distinctive, may appear within a phrase but not in the leftmost or rightmost position in the phrase. Essentially, the document is segmented at words of intermediate rarity (i.e., DF(x) between R and N-R) and what remains are considered distinctive phrases. The phrases may also optionally be segmented at the glue words to obtain additional distinctive subphrases (e.g., "United States" from the example above). Phrases must contain at least two words. The second pass also builds indexes that map from documents to their sets of distinctive phrases and subphrases and from the phrases to the documents that contain them. The indexes are built in a manner that ignores duplicates.

Unlike single words of low DF, the phrases are long enough to distinguish documents that happen to use the same vocabulary, but short enough to be common among duplicate documents.

The third pass iterates over the document index identifiers, since it is no longer necessary to use the actual documents. For each document ID, it uses the document-phrase indexes to gather a list of the phrase IDs. For each phrase ID, it iterates over document IDs

obtained from the phrase-document indexes, counting the total number of times each document ID occurs. Thus for each document ID, we have a list of document IDs of documents that overlap with the document in at least one phrase, and the number of phrases of overlap. This list of document IDs includes only those documents that have at least one phrase in common with the source document, avoiding the need to compare the source document with every other document. For each pair of documents, the number of common phrases is divided by the smaller of the number of phrases in each document. (This makes it possible to detect a small passage excerpted from a longer document.) This overlap ratio is compared with a match percentage threshold. If it exceeds the threshold the pair is reported as potential near-duplicates. These results may be either accepted as-is, or a more detailed comparison algorithm applied to the near-duplicate document pairs.

This method is rather robust, since small changes to a document have little impact on the effectiveness of the method. If there are any telltale signs of a copy left, this method will find them. Moreover, the distinctive phrases do not need to appear in the same order in the duplicate documents.

The method is also very efficient. The first two passes are linear in N. The third pass runs in time N*P, where P is the average number of documents that overlap in at least one phrase. In the worst case P is N, but typically P is R. Note that as R increases, so does the accuracy, but the running time also increases. So there is a tradeoff between running time and accuracy. In practice, however, an acceptable level of accuracy is achieved for a running time that is linear in N. This is a significant improvement over algorithms which would require pairwise comparisons of all the documents, or at least N-squared running time.

To evaluate the method's performance, I used 125 newspaper articles and their corresponding human-written summaries, for a total of 250 documents. For each pair of documents identified as near-duplicates, if the pair consisted of an article and its summary, it was counted as a correct match. Otherwise, it was counted as an incorrect match. (For the purpose of this experiment, pairs consisting of a document and itself were excluded. The method successfully matches any document with itself.) Using a minimum overlap threshold of 25% and a DF threshold of 5%, the method processed all 250 documents in 13 seconds and was able to match 232 of the 250 documents with their corresponding summary or article correctly, and none incorrectly. This represents a precision (accuracy) of 100%, a recall (coverage) of 92.8%, and a F1 score of 96.3%. (F1 is the harmonic mean of precision and recall.) Inspection of the results showed that in all the cases where the algorithm did not find a match, the highest-ranking document, although below the overlap threshold, was the correct match.

Possible variations include:

- Using different thresholds for the low frequency and glue words.

- Including sequences of mid-range DF word where the sequence itself has low DF. We tried this by collecting all low-DF bigrams and trigrams (not exactly the same as taking the longest sequence of words with a DF below a given threshold). Although this significantly increased the number of phrases, it yielded a slight decrease in accuracy. We believe that by restricting the words in a phrase to low-DF and glue words, we reduce the likelihood that the low-DF sequence occurred by chance.

- Factoring in the number of words in a phrase as a measure of the phrase's complexity in addition to rarity. For example, dividing the length of the phrase by the phrase's DF

(e.g., TL/DF or log(TL)/DF). Although this yields a preference for longer phrases, it also allows longer phrases to have higher DF and so be less distinctive.

Applications of the present invention include:

- Identifying duplicate and near-duplicate documents in a large collection of documents, such as document indexes, world wide web page catalogs and search engines, and collections of email messages.
- Detection of plagiarism and copyright infringement for text documents and text passages.
- Detecting copyright infringement of images according to image content without needing to add watermarks or otherwise modifying the image. This application involves canonicalization of the images by converting them to black and white and sampling them at several resolutions. Small overlapping tiles correspond to words in the present invention and horizontal and vertical sequences to the text fragments. A similar technique can be used for fingerprint identification and handwritten signature verification, but without requiring as much canonicalization.
- Determining the authorship of disputed works by identifying text fragments that are peculiar to each author's previous writings.
- Filing successive versions of a document or email message in the appropriate folder or directory.
- Matching a response to an email message with the original email message based on message content, as opposed to reference chains.
- Seeding a text classification or text clustering algorithm with a set of initial document clusters.
- Augmenting an information retrieval or text classification algorithm that uses terms that consist of single words with a small number of multi-word terms. Algorithms based on a bag-of-words model assume that each word appears independently. Although such algorithms can be extended to apply to word bigrams, trigrams, and so on, allowing all word n-grams of a particular length rapidly becomes computationally intractable. The present invention may be used to generate a small list of word n-grams to augment the bag-of-words index. These word n-grams are likely to distinguish documents, and so, if present in a query, can help narrow the search results considerably. (This is in contrast to methods based on word coocurrence statistics, which yield word n-grams that are rather common in the document set.)
- A method of information retrieval in which queries consist of a small excerpt or quotation from a document and the user wishes to find the documents that contain that quote or excerpt.

Nothing in the present invention restrictions the method to working just with documents. It can work with any text span, including sentences. For example, the present invention can be used to match sentences from one document with sentences from another. This can be useful in matching human-written summary sentences with sentences from the original document. Similarly, in a plagiarism detector, once the preferred implementation has found duplicate documents, the sentence-level version can be used to match sentences in the plagiarized copy with the corresponding sentences from the original document.

Another application of sentence matching is for identifying changes made to a document in a word processing application, where such changes need not retain the sentences, lines, or other text fragments of the document in the original order.

Claims

The claims for the present invention should include the following:

1) A method for identifying near-duplicate and duplicate documents in a large collection of documents without requiring that each document be compared with every other document.

2) The method described in 1, where distinctive features are used as a proxy for the full document for the purpose of identifying duplicate and near-duplicate documents.

3) The method described in 2, where the distinctive features consist of distinctive text fragments from the documents.

4) The method described in 3, where the distinctive text fragments are defined to be sequences of two or more words which appear in a limited number of documents from the document collection. The limit may be either a fixed user-selected constant, a fixed user-selected percentage, or a linear function of the square root or logarithm of the number of documents.

5) The method described in 4, where the text fragments may include 'glue words' like "of" and "the" within the text fragment (but not at either extreme of the fragment). Glue words are defined to be words that do not appear in at most a limited number of documents (i.e., words that appear in almost all of the documents).

6) The methods described in 4 and 5, where words are counted as being present in a document only if the word appears in the document at least a user-specified minimum number of times or at least a user-specified minimum frequency. (The frequency of a word in a document is the number of occurrences divided by the length of the document.)

7) The method described in 3, where the set of distinctive phrases in a document consists of the longest sequences of two or more words in the document that occur in at least two documents and in no more than a user-specified number of documents or percentage of the document collection. In other words, if a sequence matching these constraints is contained within another sequence that matches the constraints, only the longer sequence is retained as a distinctive phrase.

8) The method described in 3, where a distinctiveness score is calculated for each word n-gram of two or more words and the highest-scoring n-grams that are found in two or more documents are considered distinctive phrases.

9) The method described in 8, where the distinctiveness score is the reciprocal of the number of documents containing the phrase, the percentage of documents that do not contain the phrase, or either of these quantities multiplied by the number of words in the phrase or any monotonic function thereof.

10) A method for measuring the overlap between two documents by counting the number of features in common between the documents and dividing by the smaller of the number of features associated with each of the two documents.

11) The method described in 10, where the overlap is counted by iterating over the document IDs associated with each feature using a feature-document index where the features iterate over the features associated with a particular document using a document-feature index.

12) The methods described in 10 and 11, where the features include distinctive phrases or other text spans, such as sentences and lines.

13) The methods described above (1 through 12), where the methods are applied to sentences or other text spans, instead of entire documents.

14) The application of the methods described above to removing duplicates in document collections, including but not limited to web catalogs, search engines, and collections of email messages.

15) The application of the methods described above to detecting plagiarism and copyright infringement of text documents and other text spans.

16) The application of the methods described above to determining the authorship of a document or other text span.

17) The application of the methods described above to matching an email message with the responses to the email message, and vice versa.

18) The application of the methods described above to clustering successive versions of a document from among a collection of documents.

19) The application of the method described in 12 to comparing two documents in which the content in common need not appear in the same order in both documents, such as matching summary sentences with the document sentences to which they correspond.

20) The application of the methods described above for seeding a text classification or text clustering algorithm with a set of initial document clusters based on the subsets of documents identified as duplicate or near-duplicate.

21) The application of the methods described in 4, 5, 6, 7, 8 and 9 to text classification and other information retrieval methods, especially methods based on a bag of words model or which otherwise assume word independence, where the distinctive text fragments are added to the index set.

22) The application of the methods described above to matching images instead of text documents. Small tiles from an image are used instead of words and sequences of adjacent tiles instead of text fragments. Applications include detecting copyright infringement of images according to image content (without needing to modify the images by adding digital watermarks), fingerprint identification, and handwritten signature authentication.

23) The application of the methods described above to creating a document index suitable for efficiently finding the documents that contain a particular quote or memorable excerpt, even if the quote isn't recorded correctly in either the query or the documents.

Actual Reduction to Practice

The present invention was implemented in early June 1999 in the PERL programming language. A copy of the source code for the PERL implementation has been appended to this disclosure.

Prior Art

I was unable to find any prior art relating to finding duplicate documents from among a collection of documents.

There is some prior art concerning methods of comparing a single pair of known-to-be-similar documents to identify the differences between the documents. The Unix 'diff' program, as described in the Aho, Hopcroft, and Ullman text [1], uses an efficient algorithm for finding the longest common subsequence (LCS) between two sequences such as the lines in the two documents. The lines that are left when the LCS is removed represent the changes needed to transform one document into another. There are also other programs for comparing a pair of files, such as the Unix 'cmp' program. The Advanced Software patent [3] uses anchor points (points in common between two files) to identify differences between an original and modified version of a document.

The LCS algorithm could be used to identify whether two documents are near-duplicates by dividing the length of the LCS by the length of the shorter document to obtain a measure of document overlap. (This would represent a new algorithm not found in the prior art.) Even with such a measure of document similarity, we would still need to make N-squared pairwise comparisons to find the duplicate documents within a collection of N documents. Moreover, this document similarity measure depends on the text fragments appearing in the same order in both documents. The present invention does not require the text fragments to appear in any particular order and is more computationally efficient than this approach.

Another approach to comparing documents is to compute a checksum for each document. If two documents have the same checksum, they are likely to be identical. But comparing documents using checksums is an extremely fragile method, since even a single character change in a document yields a different checksum. The EDS patent [1] concerns using checksums to identify duplicate records. The Netmind patent [2] concerns using checksums to identify whether a region of a document has changed by comparing checksums for subdocument passages (e.g., the text between HTML tags). (The Netmind patent concerns using checksums for determining whether a single document has changed over time, not identifying other near-duplicate documents. A similar technique could not be used to make checksums less brittle for identifying near-duplicate documents because the text passages between HTML tags are likely to be too long for identifying anything other than an exact duplicate. Checksums are good at identifying whether a document is an exact duplicate, but not at identifying near-duplicates.)

Patrick Juola's work [2] concerns using the average length of matching character n-grams to identify similar documents. For each window of consecutive characters in the source document, he computes the average length of the longest matching subsequence at each position in the target document. This effectively computes, for every possible n-gram within the source document, the average length of match (counting the number of consecutive matching characters starting from the first character of the n-gram) at each position within the target document. This is equivalent to computing, for each n-gram, the sum of the TLTF scores for the n-gram and every prefix of the n-gram. His focus is on using this technique to applications involving very small training corpora, and has applied it to a variety of areas, including language identification, determining authorship of a document, and text classification. The present invention differs from Juola's work in several respects. The present invention is much more selective in its choice of n-grams, making it much more efficient than Juola's algorithm. (His algorithm looks at all possible n-grams.) The present invention also involves word n-grams, instead of character n-grams. The present invention counts the number of n-grams in common between two documents, rather than the sum of the average length of match for the n-gram and each of its prefixes. Juola's work also depends on the frequency of the n-grams within the document, while the present invention focuses on the frequency of the n-grams across documents. The former is not a measure of distinctiveness, while the latter is. Juola's work also effectively requires the n-grams and all subparts (at least the prefix subparts) to be of high frequency, while the present invention permits a mixture of very low frequency and very high frequency components in the n-grams, notwithstanding that Juola's frequencies and the present invention's frequencies measure considerably different quantities.

To summarize, the key differences between the present invention and prior art are as follows:

- None of the prior art compares more than two documents.
- None of the prior art allows the text fragments in each document to appear in different or arbitrary order.
- The present invention is much more efficient than a pairwise comparison of each pair of documents in the document collection or methods that blindly use all possible words or n-grams to compare documents.
- Unlike other information retrieval methods which seek the most common or topical words and phrases, the present invention uses words and phrases that are specific to the documents. Such idiosyncratic phrases are peculiar enough that appearing in two documents is good evidence that the two documents are strongly related.
- None of the prior art is selective in the choice of n-grams used to compare the documents.
- None of the prior art for selecting n-grams uses the frequency of the n-grams across documents, or permits a mixture of very low frequency and very high frequency components in the n-grams.

Articles

1. Aho, Alfred V., Hopcroft, John E., and Ullman, Jeffrey D., Data Structures and Algorithms, Addison-Wesley Publishing Company, April 1987, pages 189-192.

2. Juola, Patrick, What Can We Do With Small Corpora? Document Categorization Via Cross-Entropy, Proceedings of Workshop on Similarity and Categorization, 1997. http://www.mathcs.duq.edu/~juola/papers.html

Patents

1. Patent 5680611, Duplicate record detection, Electronic Data Systems, Filed September 29, 1995, Issued October 21, 1997.

2. Patent 5898836, Change-detection tool indicating degree and location of change of internet documents by comparison of cyclic-redundancy-check (CRC) signatures, Netmind Services, Inc., Files January 14, 1997, Issued April 27, 1999.

3. Patent 4807182, Apparatus and method for comparing data groups, Advanced Software Inc., Filed March 12, 1986, Issued February 21, 1989.

Conception of Invention

The invention was conceived by Mark Kantrowitz in late May 1999.

First Disclosure (Oral or Written) of Invention

The first explicit mention of the present invention appears in Kantrowitz's May 28, 1999 progress report, in which he wrote:

> ... The idea is to look for sequences of consecutive low-DF terms (DF = number of documents containing the term), allowing it to skip over certain extremely high-DF terms (like "of"). This seems to work very well.

Disclosure or Use

The invention has not been operated or displayed publicly.

The invention has not been disclosed in any articles, magazines, or papers.

On Sale

The invention has not be offered for sale or sold.

Inventorship

SUBMITTER #1:

Name:	Mark Kantrowitz
Home Address:	5503 Covode Street, Pittsburgh, PA 15217
Social Security #:	031-54-7069
Citizenship:	USA
Email Address:	mkant@jprc.com
Work Telephone:	1-412-683-8674
Home Telephone:	1-412-422-6190

Signature: ______________________________

Date: ______________________________

WITNESS #1:

WITNESSED AND UNDERSTOOD BY:

Name: ______________________________

Date: ______________________________

Print Name: ______________________________

WITNESS #2:

WITNESSED AND UNDERSTOOD BY:

Name: ______________________________

Date: ______________________________

Print Name: ______________________________